CLAIMS

What is claimed is:

1. A method for use with a computer system, the method comprising:

providing a first software program compiled to platform-independent code for execution in a first process of the computer system;

providing a second software program compiled to native code for execution in a second process of the computer system;

sending a message from the first process to the second process to request a memory buffer.

2. The method of claim 1, further comprising requesting a first memory buffer in the first process, the first memory buffer having a first address range.

3. The method of claim 2, further comprising sending a message from the first process to the second process to request a second memory buffer in the second process.

4. The method of claim 3, further comprising allocating a second address range in the second process for the second memory buffer.

5. The method of claim 4, further comprising, in the second process, generating a first identifier associated with the second address range.

6. The method of claim 5, further comprising creating the second memory buffer in the second process, the second memory buffer associated with the second address range.

7. The method of claim 6, further comprising recording information relating to the first memory buffer and to the second memory buffer.

8. The method of claim 7, further comprising sending the first identifier and a second identifier from the second process to the first process, the second identifier representing the second memory buffer.

9. The method of claim 8, further comprising mapping the first address range to a physical memory area identified by the first identifier.

10. The method of claim 9, wherein the first address range of the first process and the second address range of the second process both map to a common physical memory area.

11. A method of processing a request to create a memory buffer object for use in a computer system, the method comprising:

receiving a request to create a memory buffer object from a software program compiled to a computer system-independent language;

generating a first memory buffer object in a first process executing the software program;

generating a second memory buffer object via a second process, the second process executing native code.

12. The method of claim 11, wherein the first memory buffer object and the second memory buffer object are mapped to a common memory area shared by the first process and the second process.

13. The method of claim 11, further comprising, at the first process, receiving an identifier associated with the second memory buffer object from the second process.

14. The method of claim 13, further comprising returning the identifier to the software program.

15. The method of claim 11, wherein the computer system is adapted to execute multiple software programs, a first of the software programs coded in a different programming language than a second of the software programs.

16. A computer system including a processor and a memory, the computer system comprising:

a first process to execute a first software program coded in a safe language;

a second process to execute a second software program coded in an unsafe language; and

an inter-process communication mechanism that allows data message communication between the first process and the second process, the inter-process communication mechanism including a command that provides for transmission of a message from the first process to the second process to request creation of a direct buffer that is mapped from both the first process and the second process to a common memory area.

17. A method for use with a computer system, the method comprising:

providing a first software program compiled to platform-independent code for execution in a first process;

providing a second software program compiled to native code for execution in a second process;

requesting a first memory buffer in the first process, the first memory buffer having a first address range;

sending a message from the first process to the second process to request a second memory buffer in the second process; and

mapping the first address range to a physical memory area identified by a first identifier received from the second process.

18. A method for use with a computer system, the computer system having a first software program compiled to platform-independent code for execution in a first process and having a second software program compiled to native code for execution in a second process, the method comprising:

receiving a message at the second process that requests a memory buffer;

allocating an address range in the second process for the memory buffer; and

creating the memory buffer in the second process, the memory buffer associated with the address range.

19. The method of claim 18, further comprising generating a first identifier associated with the address range.

20. The method of claim 19, further comprising sending the first identifier from the second process to the first process.

21. The method of claim 18, further comprising recording information relating to the memory buffer.

22. A computer system including a processor and a memory, the computer system comprising:

a first process to execute a first software program coded in a safe language;

a second process to execute a second software program coded in an unsafe language;

an inter-process communication mechanism that allows data message communication between the first process and the second process;

a first memory buffer object accessible by the first and the second process; and

a second memory buffer object accessible by the first and the second process.

23. The computer system of claim 22, wherein the first memory buffer object has a first address range in the first process, the second memory buffer object has a second address range in the first process and wherein the first address range and the second address range overlap.

24. A method for use with a computer system, the method comprising:

providing a first software program compiled to platform-independent code for execution in a first process of the computer system;

providing a second software program compiled to native code for execution in a second process of the computer system; and

sending a message from the second process to the first process to request a memory buffer.

25. A method of processing a request to create a memory buffer object for use in a computer system, the method comprising:

receiving a request to create a first memory buffer object from a software program compiled to a computer system-independent language;

receiving a request to create a second memory buffer object from the software program compiled to a computer system-independent language;

allocating a first memory address range for the first memory buffer object in a first process executing the software program;

allocating a second memory address range for the second memory buffer object in a first process executing the software program, the second memory address range at least partially overlapping the first memory address range; and

allocating a memory address range for each of the first memory buffer object and the second memory buffer object, in a second process, the second process executing native code.